## Claims

What is claimed is:

1. A method of communicating over a controller area network (CAN) bus, comprising:

routing registration information from a plurality of processor-enabled peripheral devices to a controlling software component;

routing a periodic heartbeat message from the controlling software component to the plurality of processor-enabled peripheral devices to enable each of the plurality of processor-enabled peripheral devices to maintain its registered status; and

if necessary, routing messages from the controlling software component to one or more of the plurality of processor-enabled peripheral devices on a discrete basis over the CAN bus to control the one or more of the plurality of processor-enabled peripheral devices.

2. The method of claim 1, further comprising causing the controlling software component to consecutively receive frames of a multi-frame message transmitted from one of the plurality of processor-enabled peripheral devices.

3. The method of claim 1, wherein the routing of messages from the controlling software component to one or more of the plurality of processor-enabled peripheral devices on a discrete basis to control the one or more of the plurality of processor-enabled peripheral devices comprises routing messages each having a like header to the one or more of the plurality of processor-enabled peripheral devices on a discrete basis to control the one or more of the plurality of processor-enabled peripheral devices.

4. The method of claim 3, wherein the routing of messages each having a like header to one or more of the plurality of processor-enabled peripheral devices on a discrete basis to control the one or more of the plurality of processor-enabled peripheral devices comprises routing messages each having a common header component and a CAN header component to the one or more of the plurality of processor-enabled peripheral devices on a discrete basis to control the one or more of the plurality of processor-enabled peripheral devices.

5. The method of claim 4, wherein the routing of messages each having a common header component and a CAN header component to the one or more of the plurality of processor-enabled peripheral devices on a discrete basis to control the one or more of the plurality of processor-enabled peripheral devices further comprises routing messages each having a common header component and a CAN header component without specific knowledge by the controlling software component of the CAN header component.

6. A method of communicating over a controller area network (CAN) bus, comprising:

routing a registration message from a processor-enabled peripheral device to a controlling software component;

at the processor-enabled peripheral device, periodically receiving a heartbeat message from the controlling software component subsequent to the routing of a registration message from a processor-enabled peripheral device to a controlling software component; and

receiving at the processor-enabled peripheral device discrete control messages that are transmitted from the controlling software component.

7. The method of claim 6, wherein the receiving at the processor-enabled peripheral device discrete control messages that are transmitted from the controlling software component comprises filtering the transmitted control messages at the processor-enabled peripheral device to enable only the discrete control messages intended specifically for the processor-enabled peripheral device to reach the processor-enabled peripheral device.

8. The method of claim 7, wherein the filtering of the transmitted control messages at the processor-enabled peripheral device to enable only the discrete control messages intended specifically for the processor-enabled peripheral device to reach the processor-enabled peripheral device comprises filtering the transmitted control messages at the processor-enabled peripheral device via a hardware filter to determine whether the transmitted control messages are for a certain type of processor-controlled peripheral device, and filtering the transmitted control messages at the processor-enabled peripheral device via a software filter to determine processor-controlled peripheral device numbers from respective message CAN headers.

9. The method of claim 8, further comprising receiving at the processor-enabled peripheral device all message frames following the processor-enabled peripheral device type and number information subsequent to the filtering of processor-enabled peripheral device type and number information from the discrete control messages intended specifically for the processor-enabled peripheral device.

10. The method of claim 7, wherein the filtering the transmitted control messages at the processor-enabled peripheral device to enable only the discrete control messages intended specifically for the processor-enabled peripheral device to reach the processor-enabled peripheral device is invisible with respect to the controlling software component.

11. The method of claim 6, further comprising, at the processor-enabled peripheral device, consecutively receiving frames of a multi-frame discrete control message.

12. A controller area network (CAN) bus for enabling a controlling software component to communicate discretely with each of a plurality of processor-enabled peripheral devices irrespective of whether the processor-enabled peripheral devices are like devices, comprising:

a processor for routing control messages between the controlling software component and the plurality of processor-enabled peripheral devices;

a plurality of bus lines for connecting the processor to the controlling software component and the plurality of processor-enabled peripheral devices; and

the processor for enabling the control messages to be discretely transmitted from the controlling software component to one or more of the plurality of processor-enabled peripheral devices.

13. The CAN bus of claim 12, wherein the processor is programmed with a software switch for enabling the controlling software component to consecutively receive frames of a multi-frame message transmitted from one of the plurality of processor-enabled peripheral devices.

14. The CAN bus of claim 12, wherein the processor is programmed for enabling transmission of multi-frame CAN bus messages.

15. The CAN bus of claim 12, wherein the processor is further for generating a CAN header component for each of the control messages transmitted from the controlling software component to enable the control messages to be discretely transmitted from the controlling software component to one or more of the plurality of processor-enabled peripheral devices.

16. The CAN bus of claim 12, wherein the processor is further for causing frames of a multi-frame message transmitted to one of the plurality of processor-enabled peripheral devices from the controlling software component to be consecutively received at the one of the plurality of processor-enabled peripheral devices.

17. The CAN bus of claim 12, wherein the processor and the plurality of bus lines are implemented on a controlling board of a wireless base station.

18. A method of handling registration of a processor-enabled peripheral device with a controlling software component, comprising:

creating a logical connection between the processor-enabled peripheral device and the controlling software component;

routing a device registration message from the processor-enabled peripheral device to the controlling software component;

routing an acknowledgement message from the controlling software component to the processor-enabled peripheral device indicating receipt of the device registration message by the controlling software component; and

breaking the logical connection between the processor-enabled peripheral device and the controlling software component if the processor-enabled peripheral device is removed and re-introduced or if the controlling software component is reset.

19. The method of claim 18, further comprising repeating the creating of a logical connection between the processor-enabled peripheral device and the controlling software component, the routing of a device registration message from the processor-enabled peripheral device to the controlling software component, and the routing of an acknowledgement message from the controlling software component to the processor-enabled peripheral device indicating receipt of the acknowledgement message by the controlling software component subsequent to the breaking of the logical connection between the processor-enabled peripheral device and the controlling software component when the processor-enabled peripheral device is removed and re-introduced or when the controlling software component has been reset.

20. The method of claim 18, further comprising:

creating respective logical connections between a plurality of processor-enabled peripheral devices of a predetermined processor-controlled peripheral device type and the controlling software component;

routing respective device registration messages from the plurality of processor-enabled peripheral devices to the controlling software component;

routing respective acknowledgement messages from the controlling software component to the plurality of processor-enabled peripheral devices indicating receipt of the respective acknowledgement messages by the controlling software component; and

breaking the respective logical connections between the plurality of processor-enabled peripheral devices and the controlling software component if one or more of the plurality of processor-enabled peripheral devices is removed and re-introduced or if the controlling software component is reset.

21. The method of claim 18, wherein the creating of a logical connection between the processor-enabled peripheral device and the controlling software component comprises:

receiving a component registration message for a predetermined device type from the controlling software component;

receiving a device registration message from the processor-controlled peripheral device;

adding the processor-controlled peripheral device to a routing table as an available device; and

forwarding the component registration message to the processor-enabled peripheral device.

22. The method of claim 21, wherein the receiving of a component registration message for a predetermined device type from the controlling software component occurs subsequent to the receiving of a device registration message from the processor-controlled peripheral device and the adding of the processor-controlled peripheral device to a routing table as an available device.

23. A network area controller comprising:

a controlling software component for controlling one or more processor-controlled peripheral devices of a predetermined device type;

a switch for creating a logical connection between the one or more of the processor-enabled peripheral devices of the predetermined device type and the controlling software component, for routing a device registration message from each of the one or more of the processor-enabled peripheral devices of a predetermined device type to the controlling software component, and for routing an acknowledgement message from the controlling software component to each of the one or more of the processor-enabled peripheral devices of a predetermined device type indicating receipt of the device registration message from each of the one or more of the processor-enabled peripheral devices of a predetermined device type by the controlling software component; and

the switch also for breaking the logical connection between the one or more of the processor-enabled peripheral devices of a predetermined device type and the controlling software component if one of the one or more of the processor-enabled peripheral devices of a predetermined device type is removed and re-introduced or if the controlling software component is reset.

24. The network area controller of claim 23, wherein the switch is further for re-creating a logical connection between the one or more of the processor-enabled peripheral devices of the predetermined device type and the controlling software component, for routing a device registration message from each of the one or more of the processor-enabled peripheral devices to the controlling software component, and for routing an acknowledgement message from the controlling software component to each of the one or more of the processor-enabled peripheral devices indicating receipt of the device registration message from each of the one or more of the processor-enabled peripheral devices by the controlling software component subsequent to breaking the logical connection between the one or more of the processor-enabled peripheral devices and the controlling software component.

25. The network area controller of claim 23, wherein the switch is for creating a logical connection between the one or more of the processor-enabled peripheral devices of the predetermined device type and the controlling software component by:

receiving a component registration message for a predetermined device type from the controlling software component;

receiving a device registration message from each of the one or more processor-controlled peripheral devices of the predetermined device type;

adding the one or more processor-controlled peripheral devices to a routing table as available devices; and

forwarding the component registration message to the one or more processor-enabled peripheral devices.

26. The network area controller of claim 25, wherein the receiving of a component registration message for a predetermined device type from the controlling software component occurs subsequent to the receiving of a device registration message from each of the one or more processor-controlled peripheral devices and the adding of the one or more processor-controlled peripheral devices to a routing table as available devices.

27. The network area controller of claim 22, wherein the switch is further for enabling the controlling software component to dynamically configure the plurality of processor-controlled peripheral devices over a CAN bus.